

Optimisasi Pembagian Tugas Kelompok Menggunakan Algoritma Branch and Bound

Dzaki Muhammad - 13519049
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13519049@std.stei.itb.ac.id

Abstract—Dalam mengerjakan tugas kelompok, salah satu aspek yang paling penting dan didahulukan adalah membagi tugas ke setiap anggota kelompok. Namun, kadang kala pembagian tugas dilakukan tidak dengan memperhatikan kemahiran ataupun kemauan anggota sehingga tugas tidak dapat dikerjakan dengan lancar. Selain itu, juga terdapat faktor seperti kesibukan atau ketersediaan waktu anggota dalam waktu pengerjaan tugas kelompok yang dapat menghambat pekerjaan. Maka, penulis mengimplementasikan algoritma branch and bound untuk optimasi pembagian tugas kelompok.

Keywords—branch and bound, optimasi, tugas kelompok, kemahiran, kemauan, waktu

I. PENDAHULUAN

Dalam pembelajaran, baik siswa maupun mahasiswa sering dihadapi dengan berbagai tugas. Tugas yang diberikan dapat berupa tugas individu ataupun tugas kelompok. Tugas individu adalah tugas yang ditanggung jawabkan kepada satu orang sedangkan tugas kelompok adalah tugas yang ditanggung jawabkan kepada sekelompok orang.

Karena sekelompok orang bertanggung jawab untuk melakukan satu tugas tertentu, dilakukan pembagian tugas atau sub-tugas kepada masing-masing orang dengan tujuan membuat pekerjaan kelompok ini menjadi lebih ringan dan lebih efisien. Namun, pada kenyataannya pembagian tugas dalam tugas kelompok dapat menyebabkan hambatan dalam pengerjaan jika dilakukan dengan tidak baik, merata, dan anggota tidak memiliki rasa tanggung jawab terhadap tugas yang sudah dibagikan.

Permasalahan pertama adalah anggota bisa mendapat tugas atau sub-tugas yang ia tidak minati. Minat atau kemauan anggota terhadap suatu tugas menjadi satu faktor yang sangat penting dalam mendukung efisiensi pengerjaan tugas baik dalam tugas yang sudah dibagi maupun tugas kelompok secara keseluruhan. Pengerjaan tugas anggota yang tidak didukung dengan kemauan dapat berakibat kepada penundaan tugas, kemalasan.

Kemudian terdapat permasalahan mengenai kemahiran anggota dalam mengerjakan suatu tugas. Kemahiran anggota dalam mengerjakan suatu tugas berpengaruh terhadap jumlah waktu yang diperlukan oleh anggota tersebut untuk mengerjakan tugas yang sudah diberikan kepadanya. Jika anggota sudah mahir untuk mengerjakan tugas yang diberikan,

maka waktu yang diperlukan untuk mengerjakan tugas tersebut dapat menjadi lebih singkat dan pengerjaan dapat menjadi lebih efisien. Sedangkan jika anggota belum mahir untuk mengerjakan tugas yang diberikan, maka waktu yang diperlukan akan menjadi lebih banyak dan dapat menghambat efektivitas pengerjaan tugas kelompok secara keseluruhan.

Permasalahan utama lain terdapat pada ketersediaan waktu anggota selama waktu pengerjaan tugas yang diberikan. Tentunya baik siswa maupun mahasiswa memiliki berbagai aktivitas lain di luar kegiatan akademik. Adanya berbagai aktivitas ini cukup menghabiskan waktu siswa atau mahasiswa dalam mengerjakan tugas-tugas akademiknya. Ketersediaan waktu yang tidak optimal untuk mengerjakan suatu tugas yang diberikan dapat menjadi hambatan serta tantangan anggota untuk menyelesaikan tugas yang diberikan dengan optimal.

Untuk menyelesaikan permasalahan-permasalahan yang telah disebutkan, Penulis mengusulkan untuk membuat matriks nilai berdasarkan kemahiran, kemauan, dan ketersediaan waktu yang dimiliki setiap anggota terhadap masing-masing modul tugas yang telah dibagi berdasarkan jumlah anggota lalu menggunakan algoritma branch and bound untuk optimasi pembagian tugas tersebut.

II. LANDASAN TEORI

A. Algoritma Branch and Bound

Algoritma *branch and bound* adalah algoritma pencarian yang digunakan untuk berbagai persoalan yang melibatkan optimisasi, yaitu meminimalkan atau memaksimalkan suatu fungsi objektif dan tidak melanggar batasan yang diberikan oleh persoalan tersebut.

Algoritma ini dapat dikatakan sebagai gabungan dari algoritma *breadth first search* (BFS) dan *least cost search*. Perbedaan antara BFS murni dengan *branch and bound* berada pada ekspansi simpulnya, pada BFS, simpul yang akan diekspansi mengikuti aturan FIFO (*First In First Out*) sedangkan pada *branch and bound*, setiap simpul diberi sebuah nilai bobot (cost) terlebih dahulu kemudian simpul yang akan diekspansi adalah simpul yang memiliki bobot paling optimal.

Jika dibandingkan dengan algoritma *backtracking*, terdapat beberapa persamaan dan perbedaan dari algoritma *branch and bound*. Persamaan yang ada antara lain pencarian solusi dilakukan dengan membentuk pohon ruang status dan

dilakukan juga penghentian ekspansi simpul yang tidak menuju solusi yang diinginkan. Sedangkan perbedaan dari kedua algoritma ini berada pada bentuk persoalan yang diselesaikan dan pembangkitan simpul. Algoritma *backtracking* umumnya digunakan untuk persoalan non-optimasi sedangkan algoritma *branch and bound* umumnya digunakan untuk persoalan optimasi. Pembangkitan simpul pada algoritma *backtracking* umumnya dilakukan dengan metode DFS (*depth first search*) sedangkan algoritma *branch and bound* umumnya menggunakan metode *best first*.

B. Fungsi Pembatas

Sesuai dengan namanya, algoritma *branch and bound* mengimplementasikan penghentian pada simpul yang melewati batas, yaitu simpul yang tidak lagi menuju solusi yang diharapkan atau tidak mungkin menghasilkan solusi terbaik. Kriteria penghentian yang diimplementasikan dalam algoritma ini umumnya:

- Nilai bobot simpul tidak lebih baik dari nilai bobot terbaik sejauh ini
- Simpul tidak menggambarkan solusi yang *feasible* karena ada batasan yang dilanggar
- Solusi yang *feasible* pada simpul tersebut hanya terdiri atas satu titik atau tidak ada pilihan lainnya. Solusi terbaik diambil dengan membandingkan nilai fungsi objektif dengan solusi terbaik saat ini lalu mengambil nilai yang terbaik.

C. Assignment Problem

Assignment problem adalah permasalahan dalam membagikan n buah pekerjaan (*assignment*) kepada n orang. Setiap orang akan diberikan sebuah pekerjaan kemudian terdapat ongkos (*cost*) untuk menugaskan setiap orang dengan sebuah pekerjaan yang digambarkan dalam sebuah matriks nilai. Tujuan dari *assignment problem* ini adalah menugaskan n orang dengan n buah pekerjaan tersebut sehingga total ongkos yang harus dikeluarkan seminimal mungkin.

Penyelesaian dari permasalahan ini memanfaatkan algoritma *branch and bound*. Nilai batasan (*bound*) dari persoalan ini didapat dengan menjumlahkan nilai minimum dari setiap baris matriks dikarenakan tidak mungkin ada solusi yang total ongkosnya lebih kecil dari penjumlahan nilai minimum tersebut. Kemudian untuk bobot atau ongkos setiap simpul didapat dengan menjumlahkan nilai ongkos sesuai dengan status terkini dan nilai minimum pada setiap baris yang belum ditugaskan serta memperhatikan bahwa tidak ada pekerjaan yang ditugaskan ke dua orang atau lebih.

Langkah penyelesaian permasalahan ini adalah sebagai berikut. Misalkan terdapat 4 buah pekerjaan yang akan ditugaskan kepada 4 orang dan terdapat matriks ongkos sebagai berikut.

$$C = \begin{bmatrix} \text{Job 1} & \text{Job 2} & \text{Job 3} & \text{Job 4} \\ 9 & 2 & 7 & 8 \\ 6 & 4 & 3 & 7 \\ 5 & 8 & 1 & 8 \\ 7 & 6 & 9 & 4 \end{bmatrix} \begin{matrix} \text{Orang } a \\ \text{Orang } b \\ \text{Orang } c \\ \text{Orang } d \end{matrix}$$

Gambar 1. Matriks Ongkos Assignment Problem
Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf> diakses pada 8 Mei 2021

Ongkos untuk simpul akar adalah penjumlahan nilai minimum dari setiap baris matriks, $c(0) = 2 + 3 + 1 + 4 = 10$.

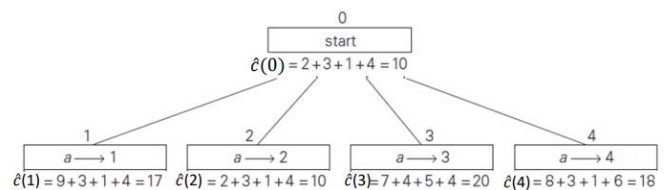
$$C = \begin{bmatrix} \text{Job 1} & \text{Job 2} & \text{Job 3} & \text{Job 4} \\ 9 & \textcircled{2} & 7 & 8 \\ 6 & 4 & \textcircled{3} & 7 \\ 5 & 8 & \textcircled{1} & 8 \\ 7 & 6 & 9 & \textcircled{4} \end{bmatrix} \begin{matrix} \text{Orang } a \\ \text{Orang } b \\ \text{Orang } c \\ \text{Orang } d \end{matrix}$$

Gambar 2. Representasi Penghitungan Ongkos Simpul Akar

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf> diakses pada 8 Mei 2021

Kemudian dari simpul akar akan dibangkitkan simpul anaknya yaitu simpul penugasan pekerjaan untuk orang a , lalu dihitung ongkos pada setiap simpul anak tersebut.

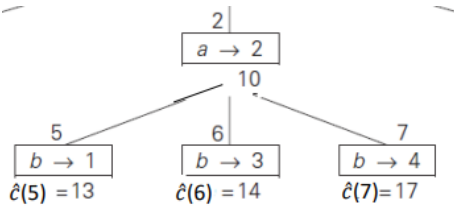


Gambar 3. Pembangkitan Anak dari Simpul Akar

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf> diakses pada 8 Mei 2021

Ongkos yang menjadi nilai minimum selain simpul akar adalah simpul 2 sehingga simpul 2 menjadi simpul yang diekspansi berikutnya.

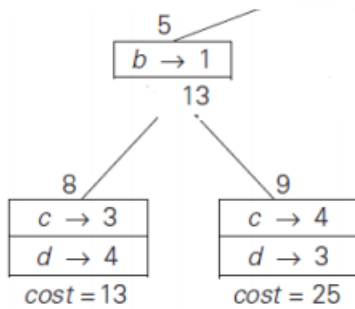


Gambar 4. Pembangkitan Anak dari Simpul 2

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf> diakses pada 8 Mei 2021

Perhatikan bahwa simpul hidup saat ini adalah simpul 1, 3, 4, 5, 6, dan 7. Simpul dengan ongkos paling kecil saat ini adalah simpul 5 sehingga simpul 5 menjadi simpul yang diekspansi berikutnya.

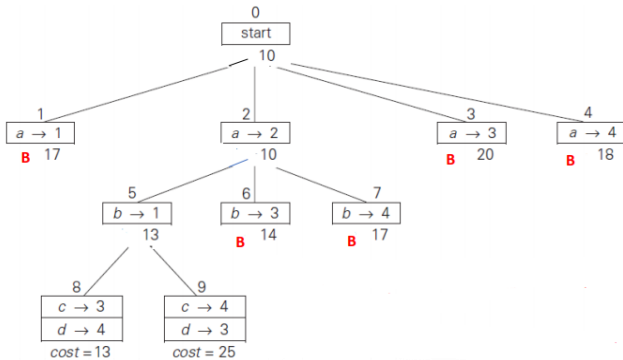


Gambar 5. Pembangkitan Anak dari Simpul 5

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf> diakses pada 8 Mei 2021

Simpul 8 adalah simpul solusi dan merupakan solusi optimal. Semua simpul hidup yang memiliki ongkos lebih besar dari 13 dimatikan karena tidak mungkin menghasilkan solusi yang lebih optimal dari 13.



Gambar 6. Pohon Ruang Status Keseluruhan

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf> diakses pada 8 Mei 2021

Solusi optimal yang diperoleh pada persoalan ini : (a → 2, b → 1, c → 3, d → 4) dengan total ongkos 13.

$$C = \begin{bmatrix} \text{Job 1} & \text{Job 2} & \text{Job 3} & \text{Job 4} \\ 9 & 2 & 7 & 8 \\ 6 & 4 & 3 & 7 \\ 5 & 8 & 1 & 4 \\ 7 & 6 & 9 & 4 \end{bmatrix} \begin{matrix} \text{Orang a} \\ \text{Orang b} \\ \text{Orang c} \\ \text{Orang d} \end{matrix}$$

$$\hat{c}(8) = 2 + 6 + 1 + 4 = 13$$

Gambar 7. Solusi Akhir dari Contoh Assignment Problem

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf> diakses pada 8 Mei 2021

D. Komponen Penting dalam Pembagian Tugas Kelompok

Hambatan yang sering terjadi dalam pengerjaan tugas adalah munculnya kemalasan dan ketidak-aktifan anggota untuk mengerjakan tugas atau sub-tugas yang sudah dibagikan dan diberikan kepada mereka. Faktor-faktor munculnya kemalasan dan ketidak-aktifan ini sangat beragam, namun umumnya sikap ini muncul akibat ketidak-mahiran anggota dalam mengerjakan tugas, minat serta kemauan anggota untuk mengerjakan tugas dengan optimal, dan juga ketersediaan waktu atau kesibukan yang anggota miliki dalam jangka waktu pengerjaan tugas. Tiga komponen ini dapat dijadikan pertimbangan ketika suatu kelompok hendak melakukan pembagian tugas sehingga setiap anggota kelompok dapat melakukan masing-masing tugasnya dengan baik.

III. OPTIMISASI PEMBAGIAN TUGAS KELOMPOK

A. Ongkos Pembagian Tugas Kelompok

Ongkos yang digunakan dalam optimisasi pembagian tugas kelompok ini adalah bobot nilai yang diperoleh berdasarkan kuesioner atau survey singkat perihal tiga komponen penting dalam pembagian tugas, yaitu kemahiran, kemauan, dan ketersediaan waktu. Sehingga nilai-nilai dari matriks ongkos yang akan digunakan adalah menggunakan rumus sebagai berikut.

$$\text{Ongkos} = (\text{Nilai kemahiran} + \text{Nilai kemauan} + \text{Nilai ketersediaan waktu}) / 3$$

B. Langkah Kerja Optimisasi Pembagian Tugas Kelompok

a) Membagi tugas kelompok menjadi modul-modul tugas berjumlah sebanyak jumlah anggota kelompok

b) Membuat kuesioner singkat mengenai kemahiran, kemauan, dan ketersediaan waktu anggota dalam mengerjakan setiap modul tugas dengan indikator skala 1-N, dengan N adalah jumlah anggota kelompok. Nilai yang lebih kecil

menunjukkan anggota lebih mahir atau lebih mau atau lebih memiliki waktu dalam mengerjakan suatu modul tugas.

c) Mengubah nilai-nilai yang diperoleh dari kuesioner menjadi nilai bobot menggunakan rumus

d) Membuat matriks nilai berdasarkan nilai bobot yang dihasilkan

e) Menerapkan algoritma branch and bound untuk menemukan pembagian tugas yang optimal menggunakan matriks nilai yang sudah dibuat

C. Notasi Algoritmik Optimisasi Pembagian Tugas Kelompok

a) Notasi algoritmik mengubah nilai kuesioner menjadi nilai bobot

```
function QuestMatrixToCostMatrix (input :
questMatrix[N][3N], output : costMatrix[N][N]) {N :
jumlah anggota kelompok}
    {inisialisasi costMatrix}
    for(i←0; i<N; i++)
        for(j←0; j<N; j++)
            costMatrix[i][j] ← 0
        endfor
    endfor

    {pembentukan costMatrix}
    for(i←0; i<N; i++)
        for(j←0; j<N; j++)
            k ← j*3
            costMatrix[i][j] ← questMatrix[i][k] +
questMatrix[i][k+1] + questMatrix[i][k+2]
        endfor
    endfor

    return costMatrix
```

b) Notasi algoritmik optimisasi pembagian tugas

Berikut merupakan Notasi algoritmik algoritma optimisasi pembagian tugas menggunakan algoritma *branch and bound* dengan referensi dari pranala <https://www.geeksforgeeks.org/job-assignment-problem-using-branch-and-bound/>.

```
procedure OptimizePembagianTugas (input:
questMatrix[N][3N]) {Prosedur mencetak
pembagian tugas yang optimal menggunakan
branch and bound}
    KAMUS
    min : Node
```

```
pq : Priority Queue of Node
child : Node

type Node : <parent : pointer to Node,
pathCost : integer,
cost : integer,
studentID : integer,
jobID : integer,
assigned : array[1..N] of boolean>

FUNGSI/PROSEDUR
function initPrioQueue(output : Priority Queue of
Node pq) {inisialisasi priority queue bertipe Node
dengan top queue merupakan Node dengan cost
terkecil}
    function QuestMatrixToCostMatrix (input :
questMatrix[N][3N], output : costMatrix[N][N])
    {Mengubah matriks nilai kuesioner menjadi matriks
nilai bobot}
        function calculateCost(costMatrix : array[1..N] of
array[1..N] of integer; sID, jID : integer; assigned :
array[1..N] of boolean) → integer {menghitung dan
memperbaharui bobot simul}
            function printAssignments(min : Node) {mencetak
pembagian tugas melalui optimisasi}
                function newNode(i, j : integer ; assigned :
array[1..N] of boolean, parent : Node) → Node
                {membuat Node baru ketika pekerjaan j ditugaskan
kepada siswa i}

                ALGORITMA
                costMatrix ←
QuestMatrixToCostMatrix(questMatrix)
                pq ← initPrioQueue()
                while (pq not empty) do
                    min ← Top(pq)
                    Pop(pq)
                    i ← min.studentID + 1

                    if (i = N) then
                        printAssignments(min)
                        return min.cost

                    for (j←0; j<N; j++)
                        if(not min.assigned[j]) then
                            child ← newNode(i, j, min.assigned, min)
                            child.pathCost ← min.pathCost +
costMatrix[i][j]
                            child.cost ← child.pathCost +
calculateCost(costMatrix, i, j, child.assigned)
                            Push(pq, child)

                REALISASI FUNGSI/PROSEDUR
                function newNode(x,y : integer ; assigned :
```

```

array[1..N] of boolean, parent : Node) → Node
for (int j = 0; j < N; j++)
    assign[j] ← assigned[j];
assign[y] = true;

node ← <parent, 0, 0, x, y, assign>

return node;

function calculateCost(costMatrix : array[1..N] of
array[1..N] of integer; sID, jID : integer; assigned :
array[1..N] of boolean) → integer
cost ← 0;

available ← {true}

for (int i ← x + 1; i < N; i++)
    min ← 9999, minIndex ← -1
    for (int j ← 0; j < N; j++)
        if (not assigned[j] and costMatrix[i][j] <
min) then
            {
                minIndex ← j;
                min ← costMatrix[i][j]
            }
        }
    cost += min;

    available[minIndex] = false
}
return cost;

function printAssignments(min : Node)
if (min.parent = NULL) then
    return
printAssignments(min.parent)
output("Siswa " + char(min.studentID + 'A') + "
mendapat tugas " + min.jobID)

```

Sumber : Dokumen Pribadi

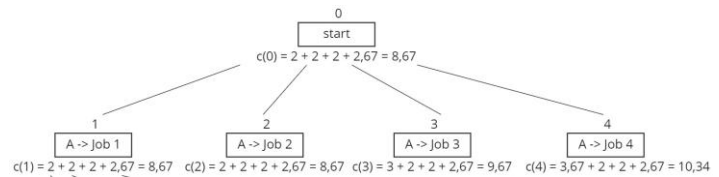
Hasil konversi matriks kuesioner menjadi matriks ongkos :

Tabel 2. Tabel Matriks Ongkos Kasus Uji

	Job 1	Job 2	Job 3	Job 4
Siswa A	2	2	3	3,67
Siswa B	2	2	3,33	2
Siswa C	2	3	2,33	2
Siswa D	2,67	3	3,67	2,67

Sumber : Dokumen Pribadi

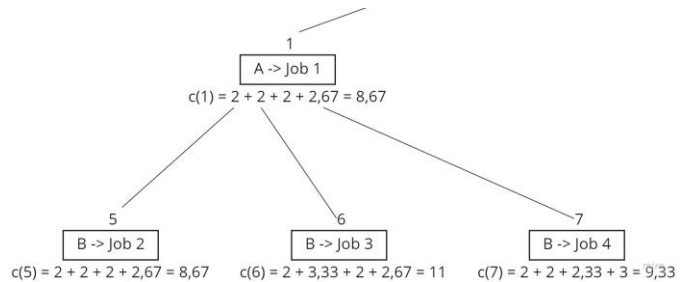
Langkah pertama sesuai dengan algoritma yang diusulkan adalah mengekspansi simpul akar sehingga terbentuk simpul-simpul anak beserta ongkosnya sebagai berikut.



Gambar 8. Ekspansi Simpul Akar Kasus Uji

Sumber : Dokumen Pribadi dibuat pada laman miro.com

Simpul berikutnya yang diekspansi adalah simpul 1 karena memiliki ongkos yang paling minimum selain simpul akar. Hasil ekspansi simpul 1 adalah sebagai berikut.



Gambar 9. Ekspansi Simpul 1 Kasus Uji

Sumber : Dokumen Pribadi dibuat pada laman miro.com

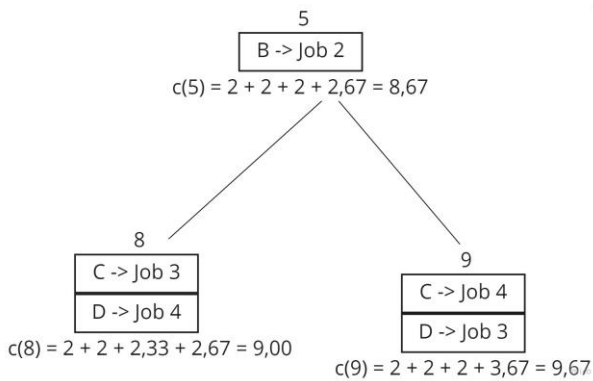
Kemudian dapat dilihat bahwa simpul 5 memiliki ongkos paling minimum sehingga menjadi simpul berikutnya yang diekspansi sehingga menghasilkan simpul-simpul anak sebagai berikut.

D. Contoh Pengujian Algoritma

Misalkan terdapat 4 modul tugas yang akan diberikan kepada 4 orang. Hasil dari kuesioner mengenai kemahiran, kemauan, dan ketersediaan waktu anggota dalam mengerjakan setiap modul tugas dinyatakan dalam tabel sebagai berikut.

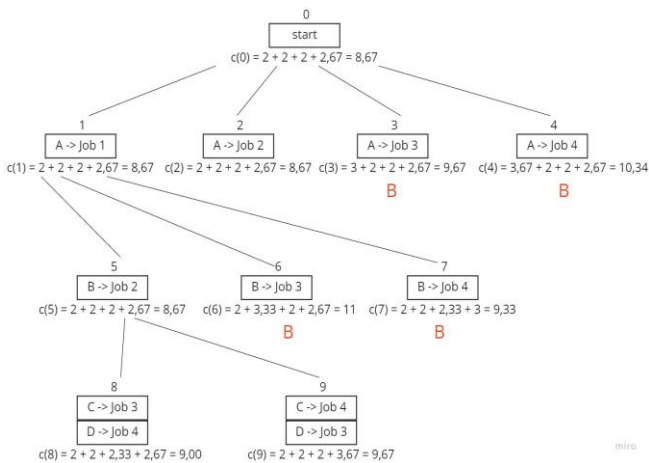
Tabel 1. Tabel Matriks Nilai Kuesioner Kasus Uji

	Job 1			Job 2			Job 3			Job 4		
	Ke ma hir an	Ke ma uan	Keter sedia an Wakt u	Ke ma hir an	Ke ma uan	Keter sedia an Wakt u	Ke ma hir an	Ke ma uan	Keter sedia an Wakt u	Ke ma hir an	Ke ma uan	Keter sedia an Wakt u
Siswa A	2	1	3	1	2	3	3	3	3	4	4	3
Siswa B	2	2	2	3	1	2	4	4	2	1	3	2
Siswa C	3	1	2	4	3	2	1	4	2	2	2	2
Siswa D	1	3	4	4	1	4	3	4	4	2	2	4



Gambar 10. Ekspansi Simpul 5 Kasus Uji
 Sumber : Dokumen Pribadi dibuat pada laman miro.com

Pencarian dihentikan karena sudah membagikan setiap tugas kepada masing-masing anggota. Pohon ruang status yang dihasilkan berdasarkan algoritma *branch and bound* yang diusulkan :



Gambar 11. Pohon Ruang Status Kasus Uji
 Sumber : Dokumen Pribadi dibuat pada laman miro.com

Berdasarkan pohon ruang status yang dihasilkan, didapat solusi optimal pertama dan solusi yang diambil yaitu : (A → Job 1, B → Job 2, C → Job 3, D → Job 4) dengan total ongkos 9,00.

IV. KESIMPULAN

Permasalahan sehari-hari namun esensial dalam menjalani kehidupan bersekolah atau perkuliahan seperti pembagian tugas dapat menjadi hambatan untuk melakukan aktivitas lainnya karena tugas berkelompok dilaksanakan dengan tidak efektif akibat pembagian tugas yang tidak optimal. Optimisasi pembagian tugas dapat dilakukan dengan menggunakan algoritma *branch and bound* seperti algoritma *assignment problem* namun dengan merubah nilai ongkos menjadi nilai yang dihasilkan dari tiga komponen penting dalam pembagian tugas yaitu kemahiran, kemauan, dan ketersediaan waktu. Pengembangan lebih lanjut dapat dilakukan dengan memberi bobot prioritas pada ketiga komponen tersebut sehingga ongkos menjadi lebih relevan dengan situasi kelompok.

PRANALA VIDEO YOUTUBE (Heading 5)

Sebagai pendukung dari pembuatan makalah ini dan untuk menambah pemahaman pembaca, penulis telah membuat video youtube yang dapat dilihat dalam tautan berikut:

<https://youtu.be/BsnaUbF2xQs>

UCAPAN TERIMA KASIH

Segala puji bagi Allah SWT yang telah memberikan penulis kelancaran sehingga dapat menyelesaikan makalah ini dengan baik dan tepat waktu. Penulis juga mengucapkan terima kasih kepada kedua orang tua serta teman-teman yang memberi dukungan selama pengerjaan makalah ini terlebih kepada teman-teman dekat penulis yang telah memberikan informasi dan referensi terkait makalah ini. Penulis menyadari masih terdapat kekurangan dan kesalahan kata dalam makalah ini, penulis berharap makalah ini dapat digunakan sebaik-baiknya dan dikembangkan sehingga lebih menghasilkan manfaat bagi masyarakat luas.

REFERENSI

- [1] A. Goel. 2018. "Job Assignment Problem using Branch And Bound". GeeksforGeeks. Diakses pada 9 Mei 2021 dari <https://www.geeksforgeeks.org/job-assignment-problem-using-branch-and-bound/>
- [2] Anonim. Diakses pada 8 Mei 2021 dari http://repository.unissula.ac.id/7203/4/BAB%20I_1.pdf
- [3] Anonim. Diakses pada 8 Mei 2021 dari <http://eprints.ums.ac.id/55978/8/BAB%20I.pdf>
- [4] R. Munir. 2021. "Algoritma Branch & Bound (Bagian 4)". Sekolah Teknik Elektro dan Informatika ITB. Diakses pada 8 Mei 2021 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf>
- [5] R. Munir, N. U. Maulidevi, M. L. Khodra. 2021. "Algoritma Branch & Bound (Bagian 1)". Sekolah Teknik Elektro dan Informatika ITB. Diakses pada 8 Mei 2021 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian1.pdf>

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Tangerang, 11 Mei 2021

Dzaki Muhammad 1351904

